

2-connecting Outerplanar Graphs without Blowing Up the Pathwidth

Jasine Babu¹, Manu Basavaraju², L. Sunil Chandran¹, and Deepak Rajendraprasad¹

¹ Indian Institute of Science, Bangalore 560012, India.

² Institute of Mathematical Sciences, Chennai 600113, India.
 jasine@csa.iisc.ernet.in, iammanu@gmail.com, {sunil,
 deepakr}@csa.iisc.ernet.in

Abstract. Given a connected outerplanar graph G with pathwidth p , we give an algorithm to add edges to G to get a supergraph of G , which is 2-vertex-connected, outerplanar and of pathwidth $O(p)$. This settles an open problem raised by Biedl [2]: as a consequence of our algorithm, we get a constant factor approximation algorithm to compute a straight line planar drawing of any outerplanar graph, with its vertices placed on a two dimensional grid of minimum height.

Keywords: Pathwidth, Outerplanar Graph, Bi-connected

1 Introduction

A graph $G(V, E)$ is outerplanar, if it has a planar embedding with all its vertices lying on the outer face. Computing planar straight line drawings of planar graphs with vertices placed on a two dimensional grid, is a well known problem in graph drawing. The height of a grid is defined as the smaller of the two dimensions of the grid. If G has a planar straight line drawing, with its vertices placed on a two dimensional grid of height h , then we call it a planar drawing of G of height h . It is known that any planar graph on n vertices can be drawn on an $(n - 1) \times (n - 1)$ sized grid [10].

We use $pw(G)$ to denote the pathwidth of a graph G . Pathwidth is a structural parameter of graphs, which is widely used in graph drawing and layout problems [2,4,12]. The study of pathwidth, in the context of graph drawings, was initiated by Dujmovic et al. [4]. It is known that any planar graph that has a planar drawing of height h has pathwidth at most h [12]. However, there exist planar graphs of constant pathwidth but requiring $\Omega(n)$ height in any planar drawing [1]. In the special case of trees, Suderman [12] showed that any tree T has a planar drawing of height at most $3pw(T) - 1$. Biedl [2] considered the same problem for the bigger class of outerplanar graphs. For any bi-connected outerplanar graph G , Biedl [2] obtained an algorithm to compute a planar drawing of G of height at most $4pw(G) - 3$. Since it is known that pathwidth is a lower bound for the height of the drawing [12], the algorithm given by Biedl

[2] is a 4-factor approximation algorithm for the problem, for any bi-connected outerplanar graph. The method in Biedl [2] is to add edges to the bi-connected outerplanar graph G to make it a maximal outerplanar graph H and then draw H on a grid of height $4pw(G) - 3$. The same method would give a constant factor approximation algorithm for arbitrary outerplanar graphs, if it is possible to add edges to any arbitrary outerplanar graph G to obtain a bi-connected outerplanar graph G' such that $pw(G') = O(pw(G))$. This was left as an open problem by Biedl [2].

Notice that, if we relax the condition that G' has to be outerplanar, this problem becomes very easy. It is easy to verify that the supergraph G' of G , obtained by making any two vertices of G universal, i.e., making them adjacent to each other and to every other vertex in the graph, has pathwidth at most $pw(G) + 2$. In this paper, we give an algorithm to augment a connected outerplanar graph G of pathwidth p by adding edges so that the resultant graph is a bi-connected outerplanar graph of pathwidth $O(p)$. The problem of augmenting outerplanar graphs to make them bi-connected, while maintaining the outerplanarity and optimizing some other properties like number of edges added [5,8] have been investigated previously.

2 Background

A *tree decomposition* of a graph $G(V, E)$ [9] is a pair (T, \mathcal{X}) , where T is a tree and $\mathcal{X} = (X_t : t \in V(T))$ is a family of subsets of $V(G)$, with the following properties:

1. $\bigcup (X_t : t \in V(T)) = V(G)$.
2. For every edge e of G there exists $t \in V(T)$ such that e has both its end points in X_t .
3. For $t, t', t'' \in V(T)$, if t' is on the path of T between t and t'' then, $X_t \cap X_{t''} \subseteq X_{t'}$.

The width of the tree decomposition is $\max_{t \in V(T)} (|X_t| - 1)$. Each $X_t \in \mathcal{X}$ is referred to as a bag in the tree decomposition. The graph G has *treewidth* w if w is the minimum such that G has a tree decomposition of width w .

A *path decomposition* (P, \mathcal{X}) of a graph G is a tree decomposition of G with the additional property that the tree P is a path. The width of the path decomposition is $\max_{t \in V(P)} (|X_t| - 1)$. The graph G has *pathwidth* w if w is the minimum such that G has a path decomposition of width w .

Without loss of generality we can assume that, in any path decomposition $(\mathcal{P}, \mathcal{X})$ of G , the vertices of the path \mathcal{P} are labeled as $1, 2, \dots$, in the order in which they appear in \mathcal{P} . Accordingly, the bags in \mathcal{X} also get indexed as $1, 2, \dots$. For each vertex $v \in V(G)$, define $FirstIndex_{\mathcal{X}}(v) = \min\{i \mid X_i \in \mathcal{X} \text{ contains } v\}$, $LastIndex_{\mathcal{X}}(v) = \max\{i \mid X_i \in \mathcal{X} \text{ contains } v\}$ and $Range_{\mathcal{X}}(v) = [FirstIndex_{\mathcal{X}}(v), LastIndex_{\mathcal{X}}(v)]$. By the definition of a path decomposition, if $t \in Range_{\mathcal{X}}(v)$, then $v \in X_t$. If v_1 and v_2 are two distinct vertices, define $Gap_{\mathcal{X}}(v_1, v_2)$ as follows:

- If $\text{Range}_{\mathcal{X}}(v_1) \cap \text{Range}_{\mathcal{X}}(v_2) \neq \emptyset$, then $\text{Gap}_{\mathcal{X}}(v_1, v_2) = \emptyset$.
- If $\text{LastIndex}_{\mathcal{X}}(v_1) < \text{FirstIndex}_{\mathcal{X}}(v_2)$, $\text{Gap}_{\mathcal{X}}(v_1, v_2)$ is $[\text{LastIndex}_{\mathcal{X}}(v_1) + 1, \text{FirstIndex}_{\mathcal{X}}(v_2)]$.
- If $\text{LastIndex}_{\mathcal{X}}(v_2) < \text{FirstIndex}_{\mathcal{X}}(v_1)$, $\text{Gap}_{\mathcal{X}}(v_1, v_2)$ is $[\text{LastIndex}_{\mathcal{X}}(v_2) + 1, \text{FirstIndex}_{\mathcal{X}}(v_1)]$.

The motivation for this definition is the following. Suppose (P, \mathcal{X}) is a path decomposition of a graph G and v_1 and v_2 are two non-adjacent vertices of G . If we add a new edge between v_1 and v_2 , a natural way to modify the path decomposition to reflect this edge addition is the following. If $\text{Gap}_{\mathcal{X}}(v_1, v_2) = \emptyset$, there is an $X_t \in \mathcal{X}$, which contains v_1 and v_2 together and hence, we need not modify the path decomposition. If $\text{LastIndex}_{\mathcal{X}}(v_1) < \text{FirstIndex}_{\mathcal{X}}(v_2)$, we insert v_1 into all $X_t \in \mathcal{X}$, such that $t \in \text{Gap}_{\mathcal{X}}(v_1, v_2)$. On the other hand, if $\text{LastIndex}_{\mathcal{X}}(v_2) < \text{FirstIndex}_{\mathcal{X}}(v_1)$, we insert v_2 to all $X_t \in \mathcal{X}$, such that $t \in \text{Gap}_{\mathcal{X}}(v_1, v_2)$. It is clear from the definition of $\text{Gap}_{\mathcal{X}}(v_1, v_2)$, that this procedure gives a path decomposition of the new graph. Whenever we add an edge (v_1, v_2) , we stick to this procedure to update the path decomposition.

A *block* of a graph G is a maximal connected subgraph of G without a cut vertex. Every block of a connected graph G is thus either a single edge which is a bridge in G , or a maximal bi-connected subgraph of G . If a block of G is not a single edge, we call it as a non-trivial block of G . Given a connected outerplanar graph G , we define a rooted tree T (hereafter referred to as the *rooted block tree* of G) as follows. The vertices of T are the blocks of G and the root of T is an arbitrary block of G which contains at least one non-cut vertex (it is easy to see that such a block always exists). Two vertices B_i and B_j of T are adjacent if the blocks B_i and B_j share a cut vertex in G . It is easy to see that T , as defined above, is a tree. In our discussions, we restrict ourselves to a fixed rooted block tree of G . If block B_i is a child block of block B_j in the rooted block tree of G , and they share a cut vertex x , we say that B_i is a child block of B_j at x .

It is known that every bi-connected outerplanar graph has a unique hamiltonian cycle [13]. Though the hamiltonian cycle of a bi-connected block of G can be traversed either clockwise or anticlockwise, let us fix one of these orderings, so that the successor and predecessor of each vertex in the hamiltonian cycle of the block is fixed. We call this order as the clockwise order. Consider a non-root block B_i of G such that B_i is a child block of its parent, at the cut vertex x . If B_i is a non-trivial block and y_i and y'_i respectively be the predecessor and successor of x in the hamiltonian cycle of B_i , we call y_i as the last vertex of B_i and y'_i as the first vertex of B_i . If B_i is a trivial block, the neighbor of x in B_i is regarded as both the first vertex and the last vertex of B_i . By the term path we always mean a simple path, i.e., a path in which no vertex repeats.

3 An overview of our method

Given a connected outerplanar graph $G(V, E)$ of pathwidth p , our algorithm will produce a bi-connected outerplanar graph $G''(V, E'')$ of pathwidth $O(p)$, where $E \subseteq E''$. Our algorithm proceeds in three stages.

(1) We use a modified version of the algorithm proposed by Govindan et al. [6] to obtain a *nice tree decomposition* (defined in Section 4) of G . Using this nice tree decomposition of G , we construct a special path decomposition of G of width at most $4p + 3$.

(2) For each cut vertex x of G , we define an ordering among the child blocks attached through x to their parent block. To define this ordering, we use the special path decomposition constructed in the first stage. This ordering helps us in constructing an outerplanar supergraph $G'(V, E')$ whose pathwidth is at most $8p + 7$, and for every cut vertex x in G' , $G' \setminus x$ has exactly two components. The properties of the special path decomposition of G obtained in the first stage is crucially used in our argument to bound the width of the path decomposition of G' , produced in the second stage.

(3) We bi-connect G' to construct $G''(V, E'')$, using a straightforward algorithm. As a by-product, this algorithm also gives us a surjective mapping from the cut vertices of G' to the edges in $E'' \setminus E'$. We give a counting argument based on this mapping and some basic properties of path decompositions to show that the width of the path decomposition of G'' produced in the third stage is at most $16p + 15$.

4 Stage 1: Construct a nice path decomposition of G

In this section, we construct a *nice tree decomposition* of the outerplanar graph G and then use it to construct a *nice path decomposition* of G . We begin by giving the definition of a nice tree decomposition.

Given an outerplanar graph G , Govindan et al. [6, Section 2] gave a linear time algorithm to construct a width 2 tree decomposition (T, \mathcal{Y}) of G where $\mathcal{Y} = (Y_t : t \in V(T))$, with the following special properties:

1. There is a bijective mapping b from $V(G)$ to $V(T)$ such that $v \in Y_{b(v)}$. (Hereafter, for any $v \in V(G)$, while referring to the vertex $b(v)$ of T , we just call it as vertex v of T .)
2. If B_i is a child block of B_j at a cut vertex x , the induced subgraph T' of T on the vertex set $V(B_i \setminus x)$ is a spanning tree of $B_i \setminus x$ and (T', \mathcal{Y}') where $\mathcal{Y}' = (Y_t : t \in V(T'))$ gives a tree decomposition of B_i .

Definition 1 (Nice tree decomposition of an outerplanar graph G). A tree decomposition (T, \mathcal{Y}) of G , where $\mathcal{Y} = (Y_t : t \in V(T))$ having properties 1 and 2 above, together with the following additional property, is called a *nice tree decomposition* of G .

3. If y_i and y'_i are respectively the last and first vertices of a non-root, non-trivial block B_i , then the bag $Y_{y_i} \in \mathcal{Y}$ contains both y_i and y'_i .

In the discussion that follows, we will show that any outerplanar graph G has a nice tree decomposition (T, \mathcal{Y}) of width at most 3. Initialize (T, \mathcal{Y}) to be the tree decomposition of G , constructed using the method proposed by Govindan

et al. [6], satisfying properties 1 and 2 of nice tree decompositions. We need to modify (T, \mathcal{Y}) in such a way that, it satisfies property 3 as well.

For every non-root, non-trivial block B_i of G , do the following. If y_i and y'_i are respectively the last and first vertices of B_i , then, for each $t \in V(B_i \setminus x)$, we insert y'_i to Y_t , if it is not already present in Y_t and we call y'_i as a *propagated* vertex.

Claim. After the modification, (T, \mathcal{Y}) remains as a tree decomposition of G .

Proof. Clearly, we only need to verify that the third property in the definition of a tree decomposition holds, for all the propagated vertices. Let y'_i be a propagated vertex, which got inserted to the bags corresponding to vertices of $B_i \setminus x$, during the modification. Let $V'_{y'_i} = \{t \mid y'_i \in Y_t, \text{ after the modification}\}$ and Let $V_{y'_i} = \{t \mid y'_i \in Y_t, \text{ before the modification}\}$. Then, clearly, $V'_{y'_i} = V_{y'_i} \cup V(B_i \setminus x)$.

Clearly, the induced subgraph of T on the vertex set $V'_{y'_i}$ is connected, since we had a tree decomposition of G before the modification. By property 2 of nice decompositions, the induced subgraph of T on the vertex set $V(B_i \setminus x)$ is also connected. Moreover, by property 1 of nice decompositions, $y'_i \in V_{y'_i}$ and hence, $y'_i \in (B_i \setminus x) \cap V_{y'_i}$. This implies that the induced subgraph of T on the vertex set $V'_{y'_i}$ is connected. \square

Claim. After the modification, (T, \mathcal{Y}) becomes a nice tree decomposition of G of width at most 3.

Proof. It is easy to verify that all the three properties required by nice decompositions are satisfied, after the modification. Moreover, for any block B_i , attached to its parent through the cut vertex x , at most one (propagated) vertex is getting newly inserted into the bags corresponding to vertices of $B_i \setminus x$. Since the size of any bag in \mathcal{Y} was at most two initially and it got increased by at most one, the new decomposition has width at most three. \square

From the claims above, we can conclude the following.

Lemma 1. *Every outerplanar graph G has a nice tree decomposition (T, \mathcal{Y}) of width 3, constructable in polynomial time.*

Definition 2 (Nice path decomposition of an outerplanar graph). *Let $(\mathcal{P}, \mathcal{X})$ be a path decomposition of an outerplanar graph G . If, for every non-root, non-trivial block B_i , there is a bag $X_t \in \mathcal{X}$ simultaneously containing both the first and last vertices of B_i , then $(\mathcal{P}, \mathcal{X})$ is called a nice path decomposition of G .*

Lemma 2. *Let G be an outerplanar graph of pathwidth p . A nice path decomposition $(\mathcal{P}, \mathcal{X})$ of G , of width at most $4p + 3$, is constructable in polynomial time.*

Proof. Let $(\mathcal{T}, \mathcal{Y})$ be a nice tree decomposition of G of width 3, obtained using Lemma 1. Obtain an optimal path decomposition $(\mathcal{P}_T, \mathcal{X}_T)$ of the tree \mathcal{T} in

polynomial time, using a standard algorithm (See [11]). Since \mathcal{T} is a spanning tree of G , the pathwidth of \mathcal{T} is at most that of G . Therefore, the width of the path decomposition $(\mathcal{P}_T, \mathcal{X}_T)$ is at most p ; i.e. there are at most $p + 1$ vertices of \mathcal{T} in each bag $X_{T_i} \in \mathcal{X}_T$.

Let $\mathcal{P} = \mathcal{P}_T$ and for each $X_{T_i} \in \mathcal{X}_T$, we define $X_i = \bigcup_{v_T \in X_{T_i}} Y_{v_T}$. Clearly,

$(\mathcal{P}, \mathcal{X})$, with $\mathcal{X} = (X_1, \dots, X_{|V(\mathcal{P}_T)|})$, is a path decomposition of G (See [6]). The width of this path decomposition is at most $4(p + 1) - 1 = 4p + 3$, since $|Y_{v_T}| \leq 4$, for each bag $Y_{v_T} \in \mathcal{Y}$ and $|X_{T_i}| \leq p + 1$, for each bag $X_{T_i} \in \mathcal{X}_T$.

Let B_i be a non-root, non-trivial block in G and y_i and y'_i respectively be the first and last vertices of B_i . Since y_i is a vertex of the tree \mathcal{T} , there is some bag $X_{T_j} \in \mathcal{X}_T$, containing y_i . The bag $Y_{y_i} \in \mathcal{Y}$ contains both y_i and y'_i , since $(\mathcal{T}, \mathcal{Y})$ is a nice tree decomposition of G . It follows from the definition of X_j , that $X_j \in \mathcal{X}$ contains both y_i and y'_i . Therefore, $(\mathcal{P}, \mathcal{X})$ is a nice path decomposition of G . \square

5 Edge addition without spoiling the outerplanarity

In this section, we prove some technical lemmas which will be later used to prove that the intermediate graph G' obtained in Stage 2 and the bi-connected graph G'' obtained in Stage 3 are outerplanar. These lemmas are easy to understand intuitively, by visualizing the outerplanar drawing. However, for completeness, we prove them combinatorially.

Lemma 3. *Let $G(V, E)$ be a connected outerplanar graph. Let u and v be two distinct non-adjacent vertices in G and let $P = (u = x_0, x_1, x_2, \dots, x_k, x_{k+1} = v)$ where $k \geq 1$ be a path in G such that:*

- P shares at most one edge with any block of G and
- for $0 \leq i \leq k$, if the block containing the edge (x_i, x_{i+1}) is non-trivial, then x_{i+1} is the successor of x_i in the hamiltonian cycle of that block.

Then the graph $G'(V, E')$, where $E' = E \cup \{(u, v)\}$, is outerplanar.

Proof. The reader may refer to Fig. 1, to get a geometric idea of statement of the lemma. It is well known that a graph G is outerplanar if and only if it contains no subgraph which is a subdivision of K_4 or $K_{2,3}$ [3]. Consider a path P between u and v as stated in the lemma.

Property 1. In every path in G from u to v , vertices x_1, \dots, x_k should appear and for $0 \leq i \leq k$, x_i should appear before x_{i+1} in any such path.

Proof. For any $1 \leq i \leq k$, the two consecutive edges $e_i = (x_{i-1}, x_i)$ and $e_{i+1} = (x_i, x_{i+1})$ of the path P belong to two different blocks of G , by assumption. Therefore, each internal vertex x_i , $1 \leq i \leq k$, is a cut vertex in G . As a result, in every path in G between u and v , vertices x_1, \dots, x_k should appear and for $0 \leq i \leq k$, x_i should appear before x_{i+1} in any such path. \square

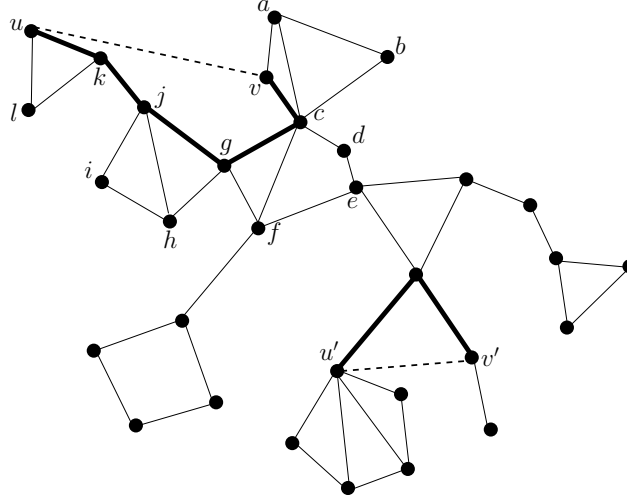


Fig. 1. The path between vertices u and v and the path between vertices u' and v' (shown in thick edges) satisfy the conditions stated in Lemma 3. By adding any one of the dotted edges (u, v) or (u', v') , the graph remains outerplanar.

Property 2. For any $0 \leq i \leq k$, there are at most two internally vertex disjoint paths in G between x_i and x_{i+1} .

Proof. Any (simple) path from (x_i, x_{i+1}) lies fully inside the block B_i , containing the edge (x_i, x_{i+1}) . If B_i is trivial, the only path from x_i to x_{i+1} is the direct edge between them. If this is not the case, B_i is bi-connected. It is a property of bi-connected outerplanar graphs, that between consecutive vertices in the hamiltonian cycle of the graph, there are exactly two internally vertex disjoint paths. By the assumption of Lemma 3, if B_i is non-trivial, then x_{i+1} is the successor of x_i in the hamiltonian cycle of B_i . Hence, the property follows. \square

We will show that if G' is not outerplanar, then G also was not outerplanar, which is a contradiction. Assume that G' is not outerplanar. This implies that there is a subgraph H' of G' that is a subdivision of K_4 or $K_{2,3}$. Since G does not have a subgraph which is a subdivision of K_4 or $K_{2,3}$, H' cannot be a subgraph of G . Hence, the new edge (u, v) should be an edge in H' and all other edges of H' are edges of G .

Case 1. H' is a subdivision of K_4 .

Let k_1, k_2, k_3 and k_4 denote the four vertices of H' , that correspond to the vertices of K_4 . We call them as branch vertices of H' . For $i, j \in \{1, 2, 3, 4\}$, $i \neq j$, let $P_{i,j}$ denote the path in H' from the branch vertex k_i to the branch vertex k_j , such that each intermediate vertex of the path is a degree two vertex in H' . Without loss of generality, assume that the edge (u, v) is part of the path $P_{1,2}$ of H' .

Claim. All of the vertices x_1, \dots, x_k appear in $P_{1,2}$. The order $<$ in which the vertices appear in $P_{1,2}$ should be one of the following three: (without loss of generality, assuming $u < v$): (1) $u < v < x_k < x_{k-1} < \dots < x_1$ (2) $x_k < x_{k-1} < \dots < x_1 < u < v$ (3) $x_j < x_{j-1} < \dots < x_1 < u < v < x_k \dots < x_{j+1}$ for some $j \in \{1, 2, \dots, k-1\}$.

Proof. Suppose x_i , $1 \leq i \leq k$, does not belong to the path $P_{1,2}$. Then, there is a path in $H' \setminus (u, v)$ between vertices u and v , avoiding the vertex x_i , since H' is a subdivision of K_4 . Since $H' \setminus (u, v)$ is a subgraph of G , this implies that there is a path in G , between u and v , which avoids x_i . This is a contradiction to Property 1. Therefore, $x_i \in P_{1,2}$. Notice that there is a path in H' between u and v which goes through the vertex k_3 . To satisfy Property 1, x_i should appear before x_{i+1} , for $0 \leq i \leq k$, in this path. Hence, one of the orderings mentioned in the claim should happen in $P_{1,2}$. \square

Let us denote the first vertex in the ordering by z_1 and the last vertex in the ordering by z_2 . (In the first case, $z_1 = u$ and $z_2 = x_1$. In the second case, $z_1 = x_k$ and $z_2 = v$. In the third case, $z_1 = x_j$ and $z_2 = x_{j+1}$.) In all the three cases of the ordering, there is a direct edge in G , between z_1 and z_2 . Notice that in any of these three possible orderings, we do not have $z_1 = u$ and $z_2 = v$ simultaneously. Since $(z_1, z_2) \neq (u, v)$, by deleting the intermediate vertices between z_1 and z_2 from the path $P_{1,2}$ and including the direct edge between z_1 and z_2 , we get a path $P'_{1,2}$ between k_1 and k_2 in G . All vertices in $P'_{1,2}$ are from the vertex set of $P_{1,2}$. Therefore, by replacing the path $P_{1,2}$ in H' by $P'_{1,2}$, we get a subgraph H of G which is a subdivision of K_4 . This means that G is not outerplanar, which is a contradiction. Therefore, H' cannot be a subdivision of K_4 .

Case 2. H' is a subdivision of $K_{2,3}$.

As earlier, let k_1, k_2, k_3, k_4 and k_5 denote the branch vertices of H' , that correspond to the vertices of $K_{2,3}$. Let k_1, k_3, k_5 be the degree 2 branch vertices in H' and k_2, k_4 be the degree 3 branch vertices of H' . For $i \in \{1, 3, 5\}$ and $j \in \{2, 4\}$, let $P_{i,j}$ denote the path in H' from vertex k_i to vertex k_j , such that each intermediate vertex of the path is a degree two vertex in H' . Also, for $i \in \{1, 3, 5\}$ and $j \in \{2, 4\}$ let $P_{j,i}$ denote the path from j to i in which the vertices in $P_{j,i}$ appear in the reverse order compared to $P_{i,j}$. Without loss of generality, assume that the edge (u, v) is part of the path $P_{1,2}$ of H' . Let $P_{4,1,2}$ denote the path in H' between vertices k_4 and k_2 , obtained by concatenating the paths $P_{4,1}$ and $P_{1,2}$.

Claim. All of the vertices x_1, \dots, x_k appear in $P_{4,1,2}$. The order $<$ in which the vertices appear in $P_{4,1,2}$ should be one of the following three (without loss of generality, assuming $u < v$): (1) $u < v < x_k < x_{k-1} < \dots < x_1$ (2) $x_k < x_{k-1} < \dots < x_1 < u < v$ (3) $x_j < x_{j-1} < \dots < x_1 < u < v < x_k \dots < x_{j+1}$ for some $j \in \{1, 2, \dots, k-1\}$.

This can be proved in a similar way as in Case 1. The remaining part of the proof is also similar. Let us denote the first vertex in the ordering by z_1 and

the last vertex in the ordering by z_2 . Repeating similar arguments as in Case 1, we can prove that by deleting the intermediate vertices between z_1 and z_2 from the path $P_{4,1,2}$ and including the direct edge between z_1 and z_2 , we get a path $P'_{4,1,2}$ between k_4 and k_2 in G . All vertices in $P'_{4,1,2}$ are from the vertex set of $P_{4,1,2}$. Therefore, by replacing the path $P_{4,1,2}$ in H' by $P'_{4,1,2}$, we get a subgraph H of G . If $P'_{4,1,2}$ has at least one intermediate vertex, the subgraph H of G , obtained by replacing the path $P_{4,1,2}$ in H' by $P'_{4,1,2}$, is a subdivision of $K_{2,3}$, where an intermediate vertex of $P'_{4,1,2}$ takes the role of the branch vertex k_1 . This contradicts the assumption that G is outerplanar.

Therefore, assume that $P'_{4,1,2}$ has no intermediate vertices, i. e., $z_1 = k_4$ and $z_2 = k_2$. In the first case of ordering mentioned in the claim above, we have $z_1 = u = x_0$ and $z_2 = x_1$. In the second case, $z_1 = x_k$ and $z_2 = v = x_{k+1}$. In the third case, $z_1 = x_j$ and $z_2 = x_{j+1}$. In each of these cases, by Property 2, there can be at most two vertex disjoint paths in G between z_1 and z_2 . But, in all these cases, there is a direct edge between z_1 and z_2 in G . Since H' is a subdivision of $K_{2,3}$, other than this direct edge, in $H' \setminus (u, v)$ there are two other paths from $z_1 = k_4$ to $z_2 = k_2$ which are internally vertex disjoint and containing at least one intermediate vertex. This will mean that there are at least three internally vertex disjoint paths from z_1 to z_2 in G , which is a contradiction. Therefore, H' cannot be a subdivision of $K_{2,3}$.

Since, G' does not contain a subgraph H' which is a subdivision of K_4 or $K_{2,3}$, G' is outerplanar. \square

Let us now analyze how the addition of an edge (u, v) , as mentioned in Lemma 3, affect the block structure and the hamiltonian cycle of each block. Assume that for each $0 \leq i \leq k$, the edge (x_i, x_{i+1}) belongs to the block B_i .

Lemma 4. – *Other than the blocks B_0 to B_k of G merging together to form a new block B' of G' , blocks in G and G' are the same.*

- *Vertices in blocks B_0 to B_k , except x_i , $0 \leq i \leq k+1$, retains their successor and predecessor in the hamiltonian cycle of B' same as it was in its respective block's hamiltonian cycle in G .*
- *Each x_i , $0 \leq i \leq k$, retains its hamiltonian cycle predecessor in B' same as it was in the block B_i of G and each x_i , $1 \leq i \leq k+1$, retains its hamiltonian cycle successor in B' same as in the block B_{i-1} of G .*

Proof. When the edge (u, v) is added, it creates a cycle containing the vertices $u = x_0, x_1, \dots, x_{k+1} = v$. Hence, the blocks B_0 to B_k of G merge together to form a single block B' in G' . It is obvious that other blocks are unaffected by this edge addition.

For simplicity, if B_i is a trivial block containing the edge (x_i, x_{i+1}) , we say that x_i and x_{i+1} are neighbors of each other in the hamiltonian cycle of B_i . For each B_i , $0 \leq i \leq k$, let $x_i, x_{i+1}, z_{i1}, z_{i2}, \dots, z_{it_i}, x_i$ be the hamiltonian cycle of B_i in G . For $0 \leq i \leq k$, let us denote the path $x_{i+1}, z_{i1}, z_{i2}, \dots, z_{it_i}$ by P_i . Then, the hamiltonian cycle of B' is $u = x_0 \circ P_k \circ P_{k-1} \circ \dots \circ P_0 \circ u$, where \circ denotes the concatenation of the paths. (For example, in Fig. 1, $u, v, a, b, c, d, e, f, g, h, i, j, k, l, u$ is the hamiltonian cycle of the new block formed, when the edge (u, v) is added.)

From this, we can conclude that the second and third parts of the lemma holds. \square

6 Stage 2: Construction of G' and its path decomposition

For each cut vertex x of G , we define an ordering among the child blocks attached through x to their parent block, using the nice path decomposition $(\mathcal{P}, \mathcal{X})$ of G obtained using Lemma 2. This ordering is then used in defining a supergraph $G'(V, E')$ of G such that for every cut vertex x in G' , $G' \setminus x$ has exactly two components. Using repeated applications of Lemma 3, we then show that G' is outerplanar. We extend the path decomposition $(\mathcal{P}, \mathcal{X})$ of G to a path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' , as described in Section 2. By a counting argument using the properties of the nice path decomposition $(\mathcal{P}, \mathcal{X})$, we show that the width of the path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' is at most $2p' + 1$, where p' is the width of $(\mathcal{P}, \mathcal{X})$.

6.1 Defining an ordering of child blocks

If $(\mathcal{P}, \mathcal{X})$ is a nice path decomposition of G , then, for each non-root block B of G , at least one bag in \mathcal{X} simultaneously contains both the first and last vertices of B .

Definition 3 (Sequence number of a non-root block). *Let $(\mathcal{P}, \mathcal{X})$ be the nice path decomposition of G obtained using Lemma 2. For each non-root block B of G , we define the sequence number of B as $\min\{i \mid X_i \in \mathcal{X} \text{ contains both the first and last vertices of } B\}$.*

For each cut vertex x , there is a unique block B^x such that B^x and its child blocks are intersecting at x . For each cut vertex x , we define an ordering among the child blocks attached at x , as follows. If B_1, \dots, B_k are the child blocks attached at x , we order them in the increasing order of their sequence numbers in $(\mathcal{P}, \mathcal{X})$. If B_i and B_j are two child blocks with the same sequence number, their relative ordering is arbitrary.

From the ordering defined, we can make some observations about the appearance of the first and last vertices of a block B_i in the path decomposition. These observations are crucially used for bounding the width of the path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' . Let B_1, \dots, B_k are the child blocks attached at a cut vertex x , occurring in that order according to the ordering we defined above. For $1 \leq i \leq k$, let y_i and y'_i respectively be the last and first vertices of B_i .

Property 3. For any $1 \leq i \leq k-1$, if $\text{Gap}_{\mathcal{X}}(y'_i, y_{i+1}) \neq \emptyset$, then $\text{Gap}_{\mathcal{X}}(y'_i, y_{i+1}) = [\text{LastIndex}_{\mathcal{X}}(y'_i) + 1, \text{FirstIndex}_{\mathcal{X}}(y_{i+1})]$ and $x \in X_t$ for all $t \in \text{Gap}_{\mathcal{X}}(y'_i, y_{i+1})$.

Proof. If $\text{Gap}_{\mathcal{X}}(y'_i, y_{i+1}) \neq \emptyset$, either $\text{LastIndex}_{\mathcal{X}}(y'_i) < \text{FirstIndex}_{\mathcal{X}}(y_{i+1})$ or $\text{LastIndex}_{\mathcal{X}}(y_{i+1}) < \text{FirstIndex}_{\mathcal{X}}(y'_i)$. The latter case will imply that, sequence number of $B_{i+1} < \text{sequence number of } B_i$, which is a contradiction.

Therefore, $LastIndex_{\mathcal{X}}(y'_i) < FirstIndex_{\mathcal{X}}(y_{i+1})$ and hence $Gap_{\mathcal{X}}(y'_i, y_{i+1}) = [LastIndex_{\mathcal{X}}(y'_i) + 1, FirstIndex_{\mathcal{X}}(y_{i+1})]$.

Since x is adjacent to y'_i and y_{i+1} , we get $FirstIndex_{\mathcal{X}}(x) \leq LastIndex_{\mathcal{X}}(y'_i)$ and $LastIndex_{\mathcal{X}}(x) \geq FirstIndex_{\mathcal{X}}(y_{i+1})$. We can conclude that $Range_{\mathcal{X}}(x) \supseteq [LastIndex_{\mathcal{X}}(y'_i), FirstIndex_{\mathcal{X}}(y_{i+1})]$ and the property follows. \square

Property 4. For any $1 \leq i < j \leq k - 1$, $Gap_{\mathcal{X}}(y'_i, y_{i+1}) \cap Gap_{\mathcal{X}}(y'_j, y_{j+1}) = \emptyset$.

Proof. We can assume that $Gap_{\mathcal{X}}(y'_i, y_{i+1}) \neq \emptyset$ and $Gap_{\mathcal{X}}(y'_j, y_{j+1}) \neq \emptyset$, since the property holds trivially otherwise. By Property 3, we get, $Gap_{\mathcal{X}}(y'_i, y_{i+1}) = [LastIndex_{\mathcal{X}}(y'_i) + 1, FirstIndex_{\mathcal{X}}(y_{i+1})]$ and $Gap_{\mathcal{X}}(y'_j, y_{j+1}) = [LastIndex_{\mathcal{X}}(y'_j) + 1, FirstIndex_{\mathcal{X}}(y_{j+1})]$. Since $i + 1 \leq j$, by the property of the ordering of blocks, we know that sequence number of $B_{i+1} \leq$ sequence number of B_j . From the definitions, we have, $FirstIndex_{\mathcal{X}}(y_{i+1}) \leq$ sequence number of $B_{i+1} \leq$ sequence number of $B_j \leq LastIndex_{\mathcal{X}}(y'_j)$ and the property follows. \square

6.2 Algorithm for constructing G' and its path decomposition

Algorithm 1: Computing the intermediate supergraph G' and its path decomposition

Input: An outerplanar graph $G(V, E)$ and a nice path decomposition $(\mathcal{P}, \mathcal{X})$ of G , the rooted block tree of G , the hamiltonian cycle of each non-trivial block of G and the first and last vertices of each non-root block of G

Output: An outerplanar supergraph $G'(V, E')$ of G such that, for every cut vertex x of G' , $G' \setminus x$ has exactly two connected components, a path decomposition $(\mathcal{P}', \mathcal{X}')$ of G'

```

1  $E' = E$ 
2  $(\mathcal{P}', \mathcal{X}') = (\mathcal{P}, \mathcal{X})$ 
3 for each cut vertex  $x \in V(G)$  do
4   Let  $B_1, \dots, B_{k_x}$ , in that order, be the child blocks attached at  $x$ , according
   to the ordering defined in Section 6.1
5   for  $i = 1$  to  $k_x - 1$  do
6     Let  $y'_i$  be the first vertex of  $B_i$  and  $y_{i+1}$  be the last vertex of  $B_{i+1}$ 
7      $E' = E' \cup \{(y'_i, y_{i+1})\}$ 
8     if  $Gap_{\mathcal{X}}(y'_i, y_{i+1}) \neq \emptyset$  then
9       for  $t \in Gap_{\mathcal{X}}(y'_i, y_{i+1})$  do  $X'_t = X'_t \cup \{y'_i\}$ 
10    end
11 end
```

We use Algorithm 1 to construct $G'(V, E')$ and a path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' . The processing of each cut vertex is done in lines 3 to 10 of Algorithm 1. While processing a cut vertex x , the algorithm adds the edges $(y'_1, y_2), (y'_2, y_3), \dots, (y'_{k_x-1}, y_{k_x})$ (as defined in the algorithm) and modifies the path decomposition, to reflect each edge addition.

Lemma 5. G' is outerplanar and for each cut vertex x of G' , $G' \setminus x$ has exactly two components.

Proof. We know that G is outerplanar to begin with. At a certain stage, let x be the cut vertex taken up by the algorithm for processing (in line 3). Assume that the graph at this stage, denoted by G_0 , is outerplanar and each cut vertex x' whose processing is completed, satisfies the condition that all the child blocks attached at x' have merged together to form a single child block attached at x' .

It is clear that the child blocks attached at a vertex x remain unchanged until x is picked up by the algorithm for processing. Let B_1, \dots, B_{k_x} , in that order, be the child blocks attached at x , according to the ordering defined in Section 6.1. Let B^x be the parent block of B_1, \dots, B_{k_x} , in the current graph G_0 . For each $1 \leq i \leq k_x$, let y'_i and y_i respectively be the first and last vertices of B_i . For $1 \leq i \leq k_x - 1$, let G_i be the graph obtained, when the algorithm has added the edges up to (y'_i, y_{i+1}) .

We will prove that the algorithm maintains the following invariants, while processing the cut vertex x , for each $0 \leq i \leq k_x - 1$:

The graph G_i is outerplanar. In G_i , the blocks B_1, \dots, B_{i+1} of G_{i-1} have merged together and formed a child block B' of B^x . The vertex y'_{i+1} is the first vertex of B' . If $i \leq k_x - 2$, blocks B_{i+2}, \dots, B_{k_x} remain the same, as in G .

By our assumption, the invariants hold for G_0 . We need to show that if the invariants hold for G_{i-1} , they hold for G_i as well. Assume that the invariants hold for G_{i-1} . This means that y'_i is the first vertex of B' and y_{i+1} is the last vertex in B_{i+1} . That is, y'_i is the successor of x in B' and y_{i+1} is the predecessor of x in B_{i+1} and the edges (y_{i+1}, x) and (x, y'_i) of the path $P_i = (y_{i+1}, x, y'_i)$ belong to two different blocks of G_{i-1} . Hence, by Lemma 3, after adding the edge (y'_i, y_{i+1}) , the resultant intermediate graph G_i is outerplanar. By Lemma 4, the blocks B' and B_{i+1} merges together to form a child block of B^x in G_i and the vertex y'_{i+1} will be the successor of x in the hamiltonian cycle of this block. Remaining blocks of G_i are the same as in G_{i-1} . Thus, all the invariants hold for G_i . It follows that the graph G_{k_x-1} is outerplanar and the blocks B_1, \dots, B_{k_x} have merged together in G_{k_x-1} to form a single child block of B^x at x .

When this processing is repeated at all cut vertices, it is clear that G' is outerplanar and for each cut vertex x of G' , $G' \setminus x$ has exactly two components. \square

Lemma 6. $(\mathcal{P}', \mathcal{X}')$ is a path decomposition of G' of width at most $8p + 7$.

Proof. Algorithm 1 initialized $(\mathcal{P}', \mathcal{X}')$ to $(\mathcal{P}, \mathcal{X})$ and modified it during each edge addition. By Property 3, we have $\text{Gap}_{\mathcal{X}}(y'_i, y_{i+1}) = [\text{LastIndex}_{\mathcal{X}}(y'_i) + 1, \text{FirstIndex}_{\mathcal{X}}(y_{i+1})]$. Hence, by the modification done in lines 8 to 9 while adding a new edge (y'_i, y_{i+1}) , $(\mathcal{P}', \mathcal{X}')$ becomes a path decomposition of the graph containing the edge (y'_i, y_{i+1}) , by the method explained in Section 2. It follows that $(\mathcal{P}', \mathcal{X}')$ is a path decomposition of G' .

Consider any $X'_t \in \mathcal{X}'$. While processing the cut vertex x , if Algorithm 1 inserts a new vertex y'_i to X'_t , to reflect the addition of a new edge (y'_i, y_{i+1}) then, $t \in \text{Gap}_{\mathcal{X}}(y'_i, y_{i+1})$. Suppose (y'_i, y_{i+1}) and (y'_j, y_{j+1}) are two new edges added

while processing the cut vertex x , where, $1 \leq i < j \leq k_x - 1$. By property 4, we know that if $t \in \text{Gap}_x(y'_i, y_{i+1})$, then, $t \notin \text{Gap}_x(y'_j, y_{j+1})$. Therefore, when the algorithm is processing a cut vertex x in lines 3 to 10, at most one vertex is newly getting inserted to X'_t . Moreover, if $t \in \text{Gap}_x(y'_i, y_{i+1})$ then, the cut vertex $x \in X_t$, by Property 3. It follows that $|X'_t| \leq |X_t| + \text{number of cut vertices in } X_t \leq 2|X_t| \leq 2(4p + 4)$. Therefore, the width of the path decomposition $(\mathcal{P}', \mathcal{X}')$ is at most $8p + 7$. \square

7 Construction of G'' and its path decomposition

In this section, we give an algorithm to add some more edges to $G'(V, E')$ so that the resultant graph $G''(V, E'')$ is bi-connected. The algorithm also extend the path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' to a path decomposition $(\mathcal{P}'', \mathcal{X}'')$ of G'' . By analyzing the way in which our algorithm is adding the new edges, we show a surjective mapping from the cut vertices of G' to the edges in $E'' \setminus E'$. A counting argument based on the surjective mapping shows that the width of the path decomposition $(\mathcal{P}'', \mathcal{X}'')$ is at most $16p + 15$. For making our presentation simpler, if a block B_i is just an edge (u, v) , we abuse the definition of a hamiltonian cycle and say that u and v are clockwise neighbors of each other in the hamiltonian cycle of B_i .

Recall that the graph G' has the property that for every cut vertex x of G' , $G' \setminus x$ has exactly two components. Since any cut vertex belongs to exactly two blocks of G , based on the rooted block tree structure of G , we call them as the parent block containing x and the child block containing x . We use $\text{child}_x(B)$ to denote the child block of the block B at the cut vertex x and $\text{parent}(B)$ to denote the parent block of the block B . For a block B , $\text{next}_B(v)$ denotes the successor of the vertex v in the hamiltonian cycle of B .

To get an intuition about our algorithm, the reader may consider it as a traversal of vertices of G' , starting from a non-cut vertex in the root block of G' and proceeding to the successor of v on reaching a non-cut vertex v . On reaching a cut vertex x , the algorithm recursively traverses the child block containing x and its descendant blocks and comes back to x to continue the traversal of the remaining graph. However, before starting the recursive traversal of the child block containing x and its descendant blocks, the algorithm sets $\text{bypass}(x) = \text{TRUE}$. (Note that, since there is only one child block attached to any cut vertex, each cut vertex is bypassed only once.) In this way, when a sequence of one or more cut vertices is bypassed, an edge is added from the vertex preceding the first bypassed vertex in the sequence to the vertex succeeding the last bypassed vertex in the sequence. The path decomposition is also modified, to reflect this edge addition. The detailed algorithm to bi-connect G' is given in Algorithm 2.

To make our later analysis easier, in the following Lemma we summarize some observations about how Algorithm 2 works.

Lemma 7. *1. Inside a block, the algorithm traverses vertices in the clockwise order of the unique hamiltonian cycle of the block.*

Algorithm 2: Computing a bi-connected outerplanar supergraph

Input: An outerplanar graph $G'(V, E')$ such that $G' \setminus x$ has exactly two connected components for every cut vertex x of G' . A path decomposition $(\mathcal{P}', \mathcal{X}')$ of G' . The rooted block tree of G' , the hamiltonian cycle of each non-trivial block of G' and the first and last vertices of each non-root block of G'

Output: A bi-connected outerplanar supergraph $G''(V, E'')$ of G' , a path decomposition $(\mathcal{P}'', \mathcal{X}'')$ of G''

```

1  $E'' = E', (\mathcal{P}'', \mathcal{X}'') = (\mathcal{P}', \mathcal{X}')$ 
2 for each vertex  $v \in V(G')$  do
3    $\text{completed}(v) = \text{FALSE}$ 
4   if  $v$  is a cut vertex then  $\text{bypass}(v) = \text{FALSE}$ 
5 end
6 Choose  $v$  to be some non-cut vertex of the root block
7  $B = \text{root block}, \text{completed}(v) = \text{TRUE}, \text{completedCount} = 1$ 
8 while  $\text{completedCount} < |V(G')|$  do
9    $v' = \text{next}_B(v)$ 
10  while  $v'$  is a cut vertex and  $\text{bypass}(v')$  is FALSE do
11     $\text{bypass}(v') = \text{TRUE}, B = \text{child}_{v'}(B), v' = \text{next}_B(v')$ 
12  end
13  if  $v'$  is a cut vertex and  $\text{bypass}(v')$  is TRUE then  $B = \text{parent}(B)$ 
14     $\text{completed}(v') = \text{TRUE}, \text{completedCount} = \text{completedCount} + 1$ 
15    if  $(v, v')$  is not an edge in  $G'$  then
16       $E'' = E'' \cup \{(v, v')\}$ 
17      if  $\text{Gap}_{\mathcal{X}'}(v, v') \neq \emptyset$  then
18        if  $\text{LastIndex}_{\mathcal{X}'}(v) < \text{FirstIndex}_{\mathcal{X}'}(v')$  then for  $t \in \text{Gap}_{\mathcal{X}'}(v, v')$  do
19           $X_t'' = X_t'' \cup \{v\}$ 
20        else if  $\text{LastIndex}_{\mathcal{X}'}(v') < \text{FirstIndex}_{\mathcal{X}'}(v)$  then for
21           $t \in \text{Gap}_{\mathcal{X}'}(v, v')$  do  $X_t'' = X_t'' \cup \{v'\}$ 
22      end
23    end
24     $v = v'$ 
25 end

```

2. When the algorithm encounters a non-cut vertex x during the traversal, it declares that x is completed.
3. The algorithm encounters a cut vertex x for the first time, while traversing the parent block containing x . Then, the algorithm bypasses x (i.e. set $\text{bypass}(x) = \text{TRUE}$) and descends to the child block containing x and start traversing the child block from the successor of x in the child block's hamiltonian cycle.
4. When the algorithm encounters a cut vertex x for a second time, the current block being traversed is the child block containing x . Then the algorithm traverses x and declare that x is completed and ascends to the parent block containing x . Then it continues the traversal of the parent block containing x , by considering the successor of x in the parent block's hamiltonian cycle.

5. At the point when the algorithm declares that a cut vertex x is completed, all vertices of the child block containing x and all its descendant blocks have been completed.
6. Every vertex is encountered at least once. Every vertex is completed and a vertex which is declared completed is never encountered again. When $\text{completedCount} = |V(G')|$, all the vertices of the graph have been completed.
7. During the traversal process, when the algorithm is bypassing a sequence of one or more cut vertices, an edge is added from the vertex preceding the first bypassed vertex in the sequence to the vertex succeeding the last bypassed vertex in the sequence and the path decomposition is modified, to reflect this edge addition.
8. Every new edge added has a sequence of bypassed vertices associated with it. Each cut vertex of G' is bypassed exactly once in our traversal and hence associated with a unique edge in $E'' \setminus E'$.

Proof. Observations 1, 2, 3, 7 and 8 are easy to see from the algorithm. Observation 4 follows from Observation 3. We give a proof of the other observations.

5. Consider any situation in the execution of the algorithm when it encounters a cut vertex x for the first time and bypasses x and descend to the child block containing x . Using an induction on the number of blocks in the subtree of the rooted block tree, consisting of the child block containing x and its descendant blocks, we can give a straightforward proof of the following.

Claim. x will be encountered for a second time and declared completed. Before that, all other vertices in the child block containing x and all its descendants blocks will be completed exactly once.

6. It is clear from the algorithm that if a cut vertex x is bypassed once, the algorithm will never again descend to the child block containing x or its descendant blocks. This fact, along with part (5) above, implies that when the clockwise travel of the hamiltonian cycle of the root block reaches the anticlockwise neighbor of the vertex from where our traversal began, and the algorithm declares it completed, all vertices of G' would have got completed exactly once. Hence CompletedCount would reach $|V(G')|$ and the algorithm halts. \square

Lemma 8. G'' is bi-connected.

Proof. We prove that G'' is bi-connected by showing that G'' does not have any cut vertices. Since G'' is a supergraph of G' , if a vertex x is not a cut vertex in G' , it will not be a cut vertex in G'' . We need to show that the cut vertices in G' become non-cut vertices in G'' . Consider a newly added edge (u, v) of G'' . Without loss of generality, assume that u was completed before v in the traversal, and (x_1, x_2, \dots, x_k) is the sequence of bypassed cut vertices associated with the edge (u, v) . When our algorithm adds the edge (u, v) , it creates the cycle $u, x_1, x_2, \dots, x_k, v, u$ in the resultant graph. Recall that, for each $1 \leq i \leq k$, $G' \setminus x_i$ had exactly two components; one containing x_{i-1} and the other containing x_{i+1} . After the addition of the edge, vertices x_{i-1} , x_i and x_{i+1} lie on a common cycle. Hence, when the edge (u, v) is added, for $1 \leq i \leq k$, x_i is no longer a cut vertex.

Since every cut vertex in G' was part of the bypass sequence associated with some edge in $E'' \setminus E'$, all of them become non-cut vertices in G'' . \square

To prove that G'' is outerplanar, we can imagine the edges in $E'' \setminus E'$ being added to G' one at a time. Our method is to repeatedly use Lemma 3 and show that after each edge addition, the resultant graph remains outerplanar. Let $\{e_i = (u_i, v_i) \mid 1 \leq i \leq m = |E'' \setminus E'|\}$ be the set of edges added by Algorithm 2. Assume that, for each $1 \leq i < m$, (u_i, v_i) was added before (u_{i+1}, v_{i+1}) and u_i was completed by Algorithm 2 before v_i . Let $(x_1^i, x_2^i, \dots, x_{k_i}^i)$, where $k_i \geq 1$, be the sequence of bypassed cut vertices and $P^i = (u_i = x_0^i, x_1^i, x_2^i, \dots, x_{k_i}^i, x_{k_i+1}^i = v_i)$ be the path, associated with the edge (u_i, v_i) in G' . Let B_j^i denote the block containing the edge (x_j^i, x_{j+1}^i) in G' . Clearly, B_0^1 is the root block of G' .

We will first note down some properties maintained by Algorithm 2.

Property 5. Vertex x_1^i is the successor of u_i in the hamiltonian cycle of the block B_0^i . Vertex $x_{k_i}^i$ is the predecessor of v_i in the hamiltonian cycle of the block $B_{k_i}^i$. For each $1 \leq j \leq k_i - 1$, x_{j+1}^i is successor of x_j^i in the hamiltonian cycle of the block B_j^i . The path P^i shares only one edge with any block of G' .

Proof. This follows directly from parts 3, 7 and 8 of Lemma 7. \square

Property 6. If $1 \leq i < j \leq m$ $u_i \neq u_j$.

Proof. It is clear that, when Algorithm 2 added the edge (u_i, v_i) (see Line 15), it had $v = u_i$ and $v' = v_i$ and the vertex $v = u_i$ was completed. After adding the edge $(v, v') = (u_i, v_i)$, the algorithm reassigns $v = v' = v_i$ in Line 21. By part 6 of Lemma 7, the algorithm will never encounter the completed vertex u_i again, and hence, v is never again set to u_i . Since $v = u_j$ when the edge (u_j, v_j) is added, we have $u_i \neq u_j$. \square

We say that a non-root block B is touched, at a stage of Algorithm 2, if at that stage, Algorithm 2 has already bypassed the cut vertex y such that B is the child block containing y . At any stage of Algorithm 2, we consider the root block of G' to be touched. It is clear that, when the algorithm bypasses a cut vertex y , the parent block containing y is already a touched block. From the definition, until the algorithm bypasses the cut vertex y , the child block containing y remain untouched. Hence, we get the following property.

Property 7. When Algorithm 2 has just finished adding the edge (u_i, v_i) , the touched blocks are precisely $\bigcup_{1 \leq j \leq i} \{B_0^j, B_1^j, \dots, B_{k_j}^j\}$.

Property 8. For each $2 \leq i \leq m$, when the algorithm has just finished adding the edge (u_{i-1}, v_{i-1}) , the block B_0^i is a touched block.

Proof. If B_0^i is the root block, the property is trivially true. Assume that this is not the case. Let y be the cut vertex such that B_0^i is the child block containing y . We know that $y \neq x_1^i$, since B_0^i is the parent block containing the cut vertex

x_1^i . Just after adding the edge (u_{i-1}, v_{i-1}) , x_1^i is the next cut vertex to be bypassed. Without bypassing y earlier, the algorithm cannot reach this situation. Therefore, y belongs to the bypassed cut vertex sequence of a previously added edge (u_j, v_j) , $1 \leq j < i$. Hence, B_0^i is a touched block when the algorithm has just finished adding the edge (u_j, v_j) . \square

Property 9. For each $2 \leq i \leq m$, the blocks $\{B_1^i, B_2^i, \dots, B_{k_i}^i\}$ remain untouched when the algorithm has just finished adding the edge (u_{i-1}, v_{i-1}) .

Proof. For $1 \leq j \leq k_i$, B_j^i is the child block containing the cut vertex x_j^i . By part 8 of Lemma 7, x_j^i is not yet bypassed when the algorithm has just finished adding the edge (u_{i-1}, v_{i-1}) . Hence, by definition, the block B_j^i remain untouched when the algorithm has just finished adding the edge (u_{i-1}, v_{i-1}) . \square

Lemma 9. G'' is outerplanar.

Proof. Let $G'_0 = G'$ and for each $1 \leq i \leq m$, let $G'_i(V, E'_i)$ be the graph obtained by assigning $E'_i = E' \cup \{(u_j, v_j) \mid 1 \leq j \leq i\}$. Let B'_0 denote the root block of G' . We will prove that Algorithm 2 maintains the following invariants for each $0 \leq i \leq m$:

- The graph G'_i is outerplanar.
- When the algorithm has just finished adding the edge (u_i, v_i) , the set of touched blocks, $\bigcup_{1 \leq j \leq i} \{B_0^j, B_1^j, \dots, B_{k_j}^j\}$, have merged together and formed a single block B'_i in G'_i and the other blocks of G' remain the same in G'_i .
- If $i < m$, x_1^{i+1} is the successor of u_{i+1} in the hamiltonian cycle of the block B'_i .

By Lemma 5, $G'_0 = G'$ is outerplanar and it is clear that the above invariants hold for G'_0 . Assume that the invariants hold for each i , where $1 \leq i < h \leq m$. Consider the case when $h = i$. Since the invariants hold for $h = i - 1$, x_1^h is the successor of u_h in the hamiltonian cycle of the block B'_{h-1} . Also, by Property 7 and Property 9, $\{B_1^h, B_2^h, \dots, B_{k_h}^h\}$ remain the same in G'_{h-1} as in G' . Therefore, the path P_h continues to satisfy the pre-conditions of Lemma 3 in G'_{h-1} (Property 5). Therefore, on addition of the edge (u_h, v_h) to G'_{h-1} , the resultant graph G'_h is outerplanar, by Lemma 3.

By Lemma 4, the blocks $\{B_1^h, B_2^h, \dots, B_{k_h}^h\}$ merges with B'_{h-1} and forms the block B'_h of G'_h . Other blocks of G'_h are same as those of G'_{h-1} (and hence of G'). Finally, we have to prove that the successor of u_{h+1} in the hamiltonian cycle of the block B'_h is x_1^{h+1} , which is the same as the successor u_{h+1} in the hamiltonian cycle of the block B_0^{h+1} in G' . To see this, note that, by Lemma 4, if v' is the successor of v in the block containing the edge (v, v') before an edge (u_j, v_j) is added, it remains so after adding this edge, if $v \neq u_j$. Since, by Property 6, $u_{h+1} \neq u_j$ for any $j < h + 1$, this implies that x_1^{h+1} is the successor u_{h+1} , in the block containing the edge (u_{h+1}, x_1^{h+1}) in G'_h . However, since the edge (u_{h+1}, x_1^{h+1}) was in the block B_0^{h+1} in G' and by Property 8, B_0^{h+1} has

merged with the block B'_h of G'_h , x_1^{h+1} is the successor u_{h+1} in the hamiltonian cycle of the block B'_h . Thus, all the invariants hold for $i = h$.

By the above argument, the invariants hold for each $1 \leq i \leq m$. Since $G'' = G'_m$ by definition, G'' is outerplanar. \square

Lemma 10. $(\mathcal{P}'', \mathcal{X}'')$ is a path decomposition of G'' of width at most $16p + 15$.

Proof. It is clear that $(\mathcal{P}'', \mathcal{X}'')$ is a path decomposition of G'' , since we constructed it using the method explained in Section 2.

For each $1 \leq i \leq m$, let S_i denote the set of cut vertices that belong to the bypassed cut vertex sequence associated with the edge $e_i = (u_i, v_i) \in E'' \setminus E'$. While adding the edge e_i , a vertex was inserted into $X'_t \in \mathcal{X}'$ only if $t \in \text{Gap}_{\mathcal{X}'}(u_i, v_i)$. We will now show that, if $t \in \text{Gap}_{\mathcal{X}'}(u_i, v_i)$, then, $X'_t \cap S_i \neq \emptyset$. Without loss of generality, assume that $\text{LastIndex}_{\mathcal{X}'}(u_i) < \text{FirstIndex}_{\mathcal{X}'}(v_i)$. Let x_1, \dots, x_k be the sequence of cut vertices bypassed while adding the edge (u_i, v_i) . Since u_i is adjacent to x_1 , both of them are together present in some bag in $X'_t \in \mathcal{X}'$, with $t \leq \text{LastIndex}_{\mathcal{X}'}(u_i)$. Similarly, since v_i is adjacent to x_k , they both are together present in some bag in $X'_t \in \mathcal{X}'$, with $t \geq \text{FirstIndex}_{\mathcal{X}'}(v_i)$. The sequence x_1, \dots, x_k is a path in G' between x_1 and x_k . Therefore, every bag in $X'_t \in \mathcal{X}'$ with $t \in \text{Gap}_{\mathcal{X}'}(u_i, v_i)$ should contain at least one of the cut vertices from the set $S_i = \{x_1, \dots, x_k\}$.

Thus, by the modification done to the path decomposition to reflect the addition of the edge e_i , the size of each bag in $X''_t \in \mathcal{X}''$ with $t \in \text{Gap}_{\mathcal{X}'}(u_i, v_i)$ increases by exactly one and in that case, $X'_t \cap S_i \neq \emptyset$. The other bags are unaffected by this modification. Therefore, for any t in the index set, $|X''_t| = |X'_t| + |\{i \mid 1 \leq i \leq m, S_i \cap X'_t \neq \emptyset\}|$. But, $|\{i \mid 1 \leq i \leq m, S_i \cap X'_t \neq \emptyset\}| \leq |X'_t|$, because $S_i \cap S_j = \emptyset$, for $1 \leq i < j \leq m$, by part 8 of Lemma 7. Therefore, for any t , $|X''_t| \leq 2|X'_t| \leq 2(8p + 8)$. Therefore, width of the path decomposition $(\mathcal{P}'', \mathcal{X}'')$ is at most $16p + 15$. \square

8 Converting a bi-connected outerplanar graph to a maximal outerplanar graph

Biedl [2] describes a way to convert a bi-connected outerplanar graph to a maximal outerplanar graph, using duals. Here we give a direct way of doing the same.

Suppose G'' is a bi-connected outerplanar graph. In an outerplanar embedding of G'' , the outer face is a hamiltonian cycle of G'' and other faces are induced cycles. If all the internal faces are 3-cycles, the graph is a maximal outerplanar graph. If G'' is not maximal, choose any internal face f of G'' , which is not a 3-cycle. Let $(\mathcal{P}'', \mathcal{X}'')$ be a path decomposition of G'' . If we ignore the vertices other than those in the boundary of f , from the bags of \mathcal{X}'' , we get a path decomposition of the induced cycle, bounding f . Since this is a path decomposition of a cycle, there is at least one bag in it, which contains at least 3 vertices x, y, z from this induced cycle, out of which there is a non-adjacent pair x, y . We can add the edge (x, y) to G'' and draw this edge inside the face f ,

so that the embedding remains outerplanar. Since x and y are already present together in a bag, $(\mathcal{P}'', \mathcal{X}'')$ is a path decomposition of the new graph as well. If the new graph is not maximal, we can repeat the above processing, until the resultant graph is maximal.

9 Efficiency

For our preprocessing, we need to compute a rooted block tree of the given outerplanar graph G and compute the hamiltonian cycles of each non-trivial block. These can be done in linear time [3,7,13]. The special tree decomposition construction in Govindan et al.[6] was also doable in linear time. Using the hamiltonian cycle of each non-trivial block, we did only a linear time modification in Section 4, to produce the nice tree decomposition (T, \mathcal{Y}) of G of width 3. An optimal path decomposition of the tree T , of total size $O(n \text{ pw}(T))$ can be computed in time $O(n \text{ pw}(T))$ [11]. The time taken is $O(n \log n)$, since outerplanar graphs have pathwidth at most $\log n$, and T was a spanning tree of the outerplanar graph G . For computing the nice path decomposition $(\mathcal{P}, \mathcal{X})$ of G in Section 4, the time spent is linear in the size of the path decomposition obtained for T , i.e. $O(n \log n)$ and the total size of $(\mathcal{P}, \mathcal{X})$ is $O(n \log n)$. Computing the FirstIndex, LastIndex and Range of vertices and the sequence number of blocks can be done in time linear in the size of the path decomposition. Since the resultant graph is outerplanar, Algorithm 1 and Algorithm 2 adds only a linear number of new edges. Since the size of each bag in the path decompositions $(\mathcal{P}', \mathcal{X}')$ of G' and $(\mathcal{P}'', \mathcal{X}'')$ of G'' are only a constant times the size of the corresponding bag in $(\mathcal{P}, \mathcal{X})$, the time taken for modifying $(\mathcal{P}, \mathcal{X})$ to obtain $(\mathcal{P}', \mathcal{X}')$ and later modifying it to $(\mathcal{P}'', \mathcal{X}'')$ takes only time linear in size of $(\mathcal{P}, \mathcal{X})$; i.e., $O(n \log n)$ time. Hence, the time spent in constructing G'' and its path decomposition of width $O(\text{pw}(G))$ is $O(n \log n)$.

10 Conclusion

In this paper, we have described a $O(n \log n)$ time algorithm to add edges to a given outerplanar graph G of pathwidth p to get a bi-connected outerplanar graph G'' of pathwidth at most $16p + 15$. We also get the corresponding path decomposition of G'' in $O(n \log n)$ time. Our technique is to produce a nice path decomposition of G and make use of the properties of this decomposition, while adding the new edges. Our algorithm can be used as a preprocessing step, in the algorithm proposed by Biedl [2], to produce a planar drawing of G on a grid of height $O(p)$. As explained by Biedl [2], this is a constant factor approximation algorithm, to get a planar drawing of G of minimum height.

References

1. Biedl, T.: Small drawings of outerplanar graphs, series-parallel graphs, and other planar graphs. Discrete Comput. Geom. 45(1), 141–160 (2011)

2. Biedl, T.: A 4-approximation for the height of 2-connected outer-planar graph drawings. WAOA 2012 (2012)
3. Chartrand, G., Harary, F.: Planar permutation graphs. *Annales de l'institut Henri Poincaré (B) Probabilités et Statistiques* 3, 433–438 (1967)
4. Dujmovic, V., Morin, P., Wood, D.R.: Path-width and three-dimensional straight-line grid drawings of graphs. In: *Revised Papers from the 10th International Symposium on Graph Drawing*, pp. 42–53. GD '02, Springer-Verlag, London, UK (2002)
5. Garca, A., Hurtado, F., Noy, M., Tejel, J.: Augmenting the connectivity of outer-planar graphs. *Algorithmica* 56(2), 160–179 (2010)
6. Govindan, R., Langston, M.A., Yan, X.: Approximating the pathwidth of outer-planar graphs. *Inf. Process. Lett.* 68(1), 17–23 (1998)
7. Hopcroft, J., Tarjan, R.: Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM* 16(6), 372–378 (1973)
8. Kant, G.: Augmenting outerplanar graphs. *Journal of Algorithms* 21(1), 1–25 (1996)
9. Robertson, N., Seymour, P.D.: Graph minors. iii. planar tree-width. *J. Comb. Theory, Ser. B* 36(1), 49–64 (1984)
10. Schnyder, W.: Embedding planar graphs on the grid. In: *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pp. 138–148. SODA '90 (1990)
11. Skodinis, K.: Construction of linear tree-layouts which are optimal with respect to vertex separation in linear time. *J. Algorithms* 47(1), 40–59 (2003)
12. Suderman, M.: Pathwidth and layered drawings of trees. *Int. J. Comput. Geometry Appl.* 14(3), 203–225 (2004)
13. Syslo, M.M.: Characterizations of outerplanar graphs. *Discrete Mathematics* 26(1), 47–53 (1979)